MOODY'S RATINGS

SECTOR IN-DEPTH
18 March 2026

Send Your Feedback

TABLE OF CONTENTS

Analyst Contacts

Leroy Terrelonge +33.1.5330.5989
VP-Sr Credit Officer
leroy.terrelonge@moodys.com

Yoni Katz +1.212.553.3492
Lead Cyber Credit Risk Associate
yoni.katz@moodys.com

Sanja Nedic +33.6.8393.3742
AVP-Data Scientist
sanja.nedic@moodys.com

Lesley Ritter +1.212.553.1607
SVP-Cyber Credit Risk
lesley.ritter@moodys.com

Fabian Astic +1.212.553.6814
Managing Director, Global Head of Digital Economy
fabian.astic@moodys.com

Cybersecurity – Global

AI is uncovering software bugs faster than firms can patch them

Summary

New artificial intelligence (AI) tools are becoming powerful assistants in identifying weaknesses in software coding that hackers can exploit for unauthorized network access. While this is a positive development for software security, in practice it has meant that some corporate cybersecurity teams are struggling to keep up. At a time when criminals are accelerating their attacks on identified software vulnerabilities, an inability to patch them quickly enough leaves enterprises increasingly exposed to cyber breaches and the operational damage and disruption they can cause.

» **Software vulnerabilities are a prime vector for unauthorized network access.** Today's complex and widely reused codebases often contain human errors that attackers can exploit at scale. Because many breaches begin by leveraging known but unpatched software flaws, minimizing vulnerabilities and reducing exposure time would reduce both the likelihood and, in many cases, the severity of cyber incidents.

» **AI shows significant promise in identifying software flaws.** AI tools have uncovered previously unknown bugs, even in software that has undergone extensive security testing. They are gaining the ability to do so in an increasingly autonomous fashion.

» **But low-quality AI-generated bug reports are wasting valuable remediation time.** One downside is that a lack of human review is resulting in low-quality AI software checks that frequently produce false findings. These can overwhelm security teams and distract them from genuine risks. In response, some companies are scaling back their bug bounty programs[1], which are increasingly polluted by low-quality AI-generated submissions, likely diminishing their effectiveness in surfacing bugs.

» **The gap between time taken to exploit a bug and time taken to patch it is widening.** The growing volume of disclosed vulnerabilities leaves bugs unaddressed for extended periods. At the same time, attackers are exploiting vulnerabilities more quickly. This asymmetry increases the risk of cyberattacks.

» **More secure coding practices will reduce the growing vulnerability.** A move to address security issues earlier in the software design and development process will prevent vulnerabilities from reaching production. This will reduce both the need to patch after the software is deployed and exposure to cyberattack. AI-enabled tools will support this shift by helping developers identify and fix security issues as code is written.

Software bugs are a prime vector for unauthorized network access

Software vulnerabilities are a leading cause of cyber breaches. This is because human programmers introduce unintended errors as they develop or upgrade software, often as a result of complexity or pressure to release products quickly. These errors are often widely replicated across modern, software-dependent enterprise environments. As software becomes more intricate, tracking every scenario becomes impossible.

Cybercriminals routinely exploit known vulnerabilities as a reliable way to breach corporate IT networks. Rapid patching of identified software bugs is therefore central to lowering cyber risk.

How vulnerabilities creep into software

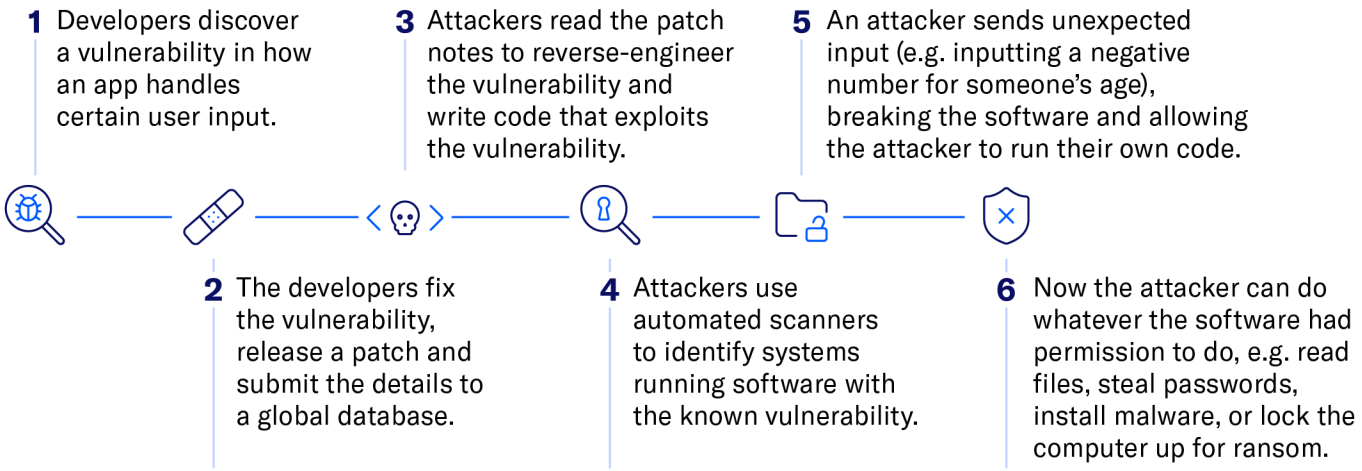
Enterprise software systems are highly complex, often comprising millions of lines of code, numerous third-party components and multiple programming languages. They operate across distributed components such as APIs and cloud services. These systems must also interact with a diverse array of operating systems, browsers and hardware, with each layer increasing the potential for vulnerabilities.

In addition, security is often not the primary design objective. Software is commonly developed with an emphasis on speed to market, functionality, useability, performance, scalability, and cost, with security addressed later. When security is added after the fact, it can be poorly integrated and less effective, leaving products more exposed than intended.

Vulnerabilities also result from trade-offs, including weak default settings for user convenience, continued support for insecure legacy systems, and performance optimization that reduces safety checks. Ongoing software changes—such as new features, cloud migrations and system updates—further increase risk by continuously introducing new coding configurations.

Software has traditionally been written by humans and is therefore inherently imperfect. Even well-resourced development teams routinely introduce flaws as they translate business logic into executable instructions, particularly as applications grow larger, more modular and more interconnected. Many of these weaknesses arise from predictable forms of human error. Developers make assumptions about how software will be used that do not hold in real-world environments. They often overlook unusual or unexpected user input, or misunderstand how software components interact. These imperfections can allow attackers to hijack the code and the systems on which they run and insert their own malware (Exhibit 1).

Exhibit 1
How attackers exploit a software vulnerability

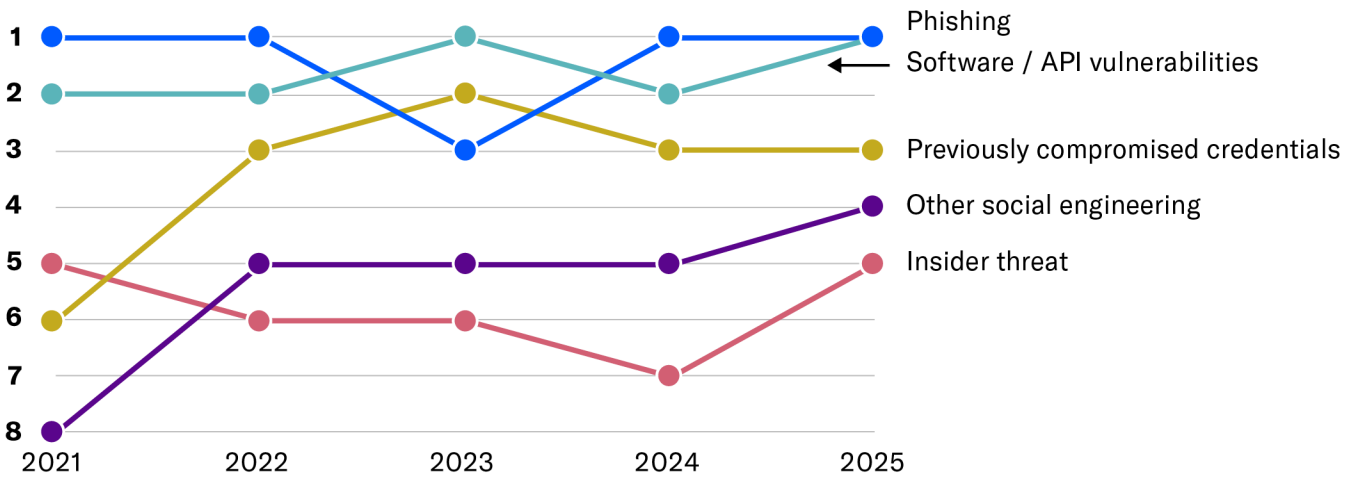


The above is one of several plausible exploitation scenarios.
Source: Moody's Ratings

Insecure code is frequently sampled and reused across systems, allowing the same flaw to appear in many places. And while artificial intelligence tools are increasingly used to automate programming, they are trained on existing human- and machine-written software and can reproduce the same weaknesses rather than eliminating them entirely. Research from Veracode, an application security company, found that, on average, AI models introduce a known security flaw in 45% of coding tasks.

Cybersecurity firms consistently find that exploiting software vulnerabilities is one of the most common ways attackers break into networks. Google's research shows that vulnerability exploitation was the single most frequent initial access method observed in 2024, responsible for roughly a third of confirmed intrusions. Palo Alto Networks' Unit 42 research similarly shows vulnerability exploitation competing with phishing as the leading initial access vector in recent years, alternating between first and second place (Exhibit 2).

Exhibit 2
Software vulnerabilities are consistently a top entry point for cybercriminals
Leading initial access vectors for cyberattackers, 2021-2025



Source: Palo Alto Networks' Unit 42

Because many cyber incidents begin with the exploitation of known software bugs, reducing the number of vulnerabilities in production software, and shortening the time they remain exposed, materially lowers the frequency and, in many cases, the severity of cyber incidents.

AI shows significant promise in identifying software vulnerabilities

As AI algorithms have advanced, they are increasingly being used as powerful tools to identify vulnerabilities in software code before they can be exploited. Large language models (LLMs), in particular, are emerging as robust assistants for software developers and security researchers by accelerating and enhancing long-standing security testing techniques.

One application is **static code analysis**. In this approach, AI tools review software code line by line, without running it, to identify known vulnerable patterns, logical inconsistencies or insecure coding practices, and to suggest safer alternatives. These tools suffer from high false-positive and false-negative rates, necessitating human review, and are therefore not suitable for full automation. They are, however, an increasingly valuable support for developers over manual efforts and simple rule-based scanners.

AI is also enhancing **software "fuzzing,"** a long-standing technique used to uncover hidden vulnerabilities. Software is often written with assumptions about how users will interact with it. When inputs deviate from those assumptions—through malformed, unexpected or extreme data—the software may behave incorrectly, potentially exposing systems or sensitive information. Fuzzing systematically tests these edge cases by feeding software large volumes of unexpected inputs to observe whether crashes or bugs occur. AI-assisted fuzzing improves this process by generating more targeted and creative test cases than traditional methods. In November 2024, Google reported that it had used LLMs to identify previously undiscovered vulnerabilities in code that had already undergone hundreds of thousands of hours of conventional fuzzing, including one vulnerability that had likely gone undetected for roughly two decades.

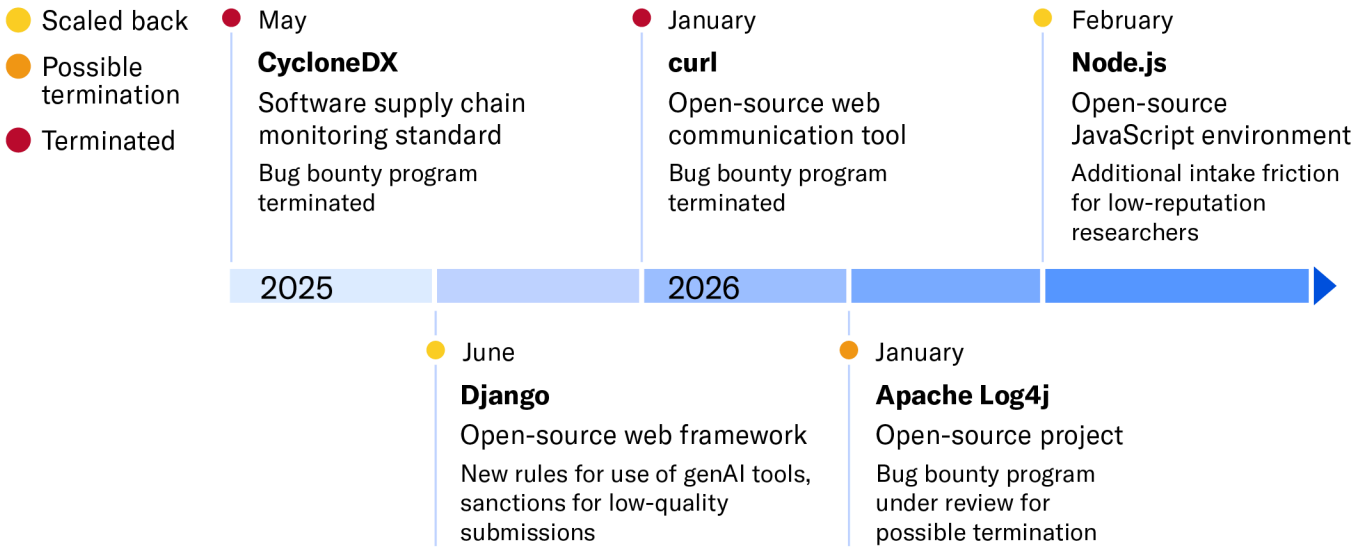
More recently, **agentic and autonomous AI systems** have begun to push vulnerability discovery further. These AI "bug hunters" can independently explore large codebases, form hypotheses about potential classes of vulnerabilities, generate proof-of-concept exploit code, and automatically validate their findings. In January, Anthropic reported that its newly released AI model, Claude Opus 4.6, identified more than 500 previously unknown high-severity vulnerabilities in widely used open-source software libraries, with minimal human prompting. Each finding was subsequently validated by Anthropic's internal security team or by external security researchers.

Low-quality AI-generated vulnerability reports are wasting valuable remediation resources

Despite advances in AI-driven security tools, human oversight remains essential. AI has a documented tendency to produce false positives and false negatives, and confidently stated but incorrect findings. Without human validation, organizations risk being overwhelmed by low-quality vulnerability reports, which can distract security teams and impair their ability to prioritize genuinely material threats.

This risk has already had practical consequences. In January, the developers of curl, one of the most widely used tools for automated web communication, announced the termination of their bug bounty program, citing an influx of low-quality, AI-generated submissions. According to the cURL project founder and lead developer, Daniel Stenberg, prior to 2025, roughly 15% of reports led to confirmed vulnerabilities. But after 2025, that figure fell to around 5%, meaning fewer than one in 20 submissions identified a real issue. The project's developers decided that the effort required to check false reports had become an inefficient use of scarce resources. This trend (see Exhibit 3) is concerning because paring back or eliminating bug bounty and vulnerability disclosure programs removes an important channel through which organizations can become aware of legitimate security weaknesses.

Exhibit 3
Low-quality AI reports are causing bug bounty programs to scale back
Status of selected bug bounty programs



Source: CycloneDX, Socket, Daniel Stenberg, Node.js, Moody's Ratings

Stenberg has credited AI-assisted tools with helping experienced contributors on his team identify and remediate 100+ genuine security issues, underscoring that the main challenge is not use of AI itself, but rather the low-quality output that occurs when that output is not validated by humans with sufficient expertise.

Similar dynamics have emerged outside formal vulnerability disclosure and bug bounty programs. In December, purportedly functional exploit code circulated widely for a vulnerability with the highest possible severity score – with a simple web request, an attacker could run whatever code they want on the affected system. The exploit code ultimately proved ineffective. Despite this, it was cited and incorporated into public reporting and vulnerability aggregators, consuming significant time across multiple security teams. The code continues to circulate online, perpetuating confusion.

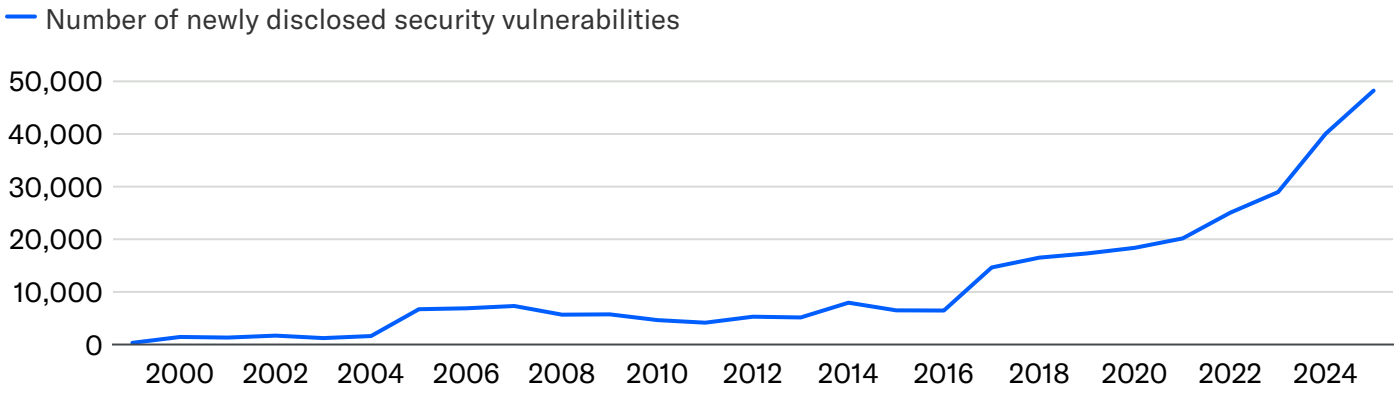
These examples highlight that the need for expert human review is a key limitation of AI-enabled security tooling. While accuracy may improve over time, the marginal cost of validating AI-generated findings is not likely to disappear.

The gap between time taken to exploit and time taken to patch is widening

A widening gap between the speed at which vulnerabilities are exploited and the time it takes organizations to remediate them is increasing cyber risk. In many cases, attackers are now able to weaponize newly disclosed software vulnerabilities well before companies can apply patches or deploy effective controls. This imbalance increases the likelihood that vulnerabilities translate into real-world cyber incidents.

One contributing factor is volume. Security teams are increasingly unable to keep pace with the sheer number of vulnerabilities identified across modern software environments. A look at the official database of published vulnerability records shows that the number of disclosed vulnerabilities has skyrocketed over the past decade, following moderate growth in the prior 15 years (Exhibit 4). The problem is likely to worsen as AI-assisted "vibe coding" enables the rapid production of large volumes of software—often without explicit security requirements or thorough review and increasingly by individuals with little or no coding experience. This will expand both the attack surface and the prevalence of vulnerabilities. As vulnerability disclosure accelerates, defenders face growing backlogs, forcing difficult prioritization decisions and leaving some weaknesses exposed for extended periods.

Exhibit 4
The pace of security vulnerability discoveries has accelerated significantly in recent years
Newly disclosed security vulnerabilities by year, 1999 - 2025



Source: The Mitre Corporation

The KEV Catalog is helping organizations reduce risk

To help address the widening gap between the growing volume of disclosed vulnerabilities and organizations' capacity to remediate them, CISA launched the Known Exploited Vulnerabilities (KEV) Catalog in 2021 to help organizations prioritize remediation of cyber risks that pose an immediate and demonstrable threat. The KEV Catalog focuses exclusively on vulnerabilities for which credible evidence of real-world exploitation exists, and is important from a credit perspective given its focus on vulnerabilities commonly used in ransomware and other disruptive attacks that can damage creditworthiness.

As seen in the exhibits below, the KEV framework is having the desired effect, with organizations fixing KEVs faster than other vulnerabilities and fixing KEVs used in ransomware attacks fastest of all.

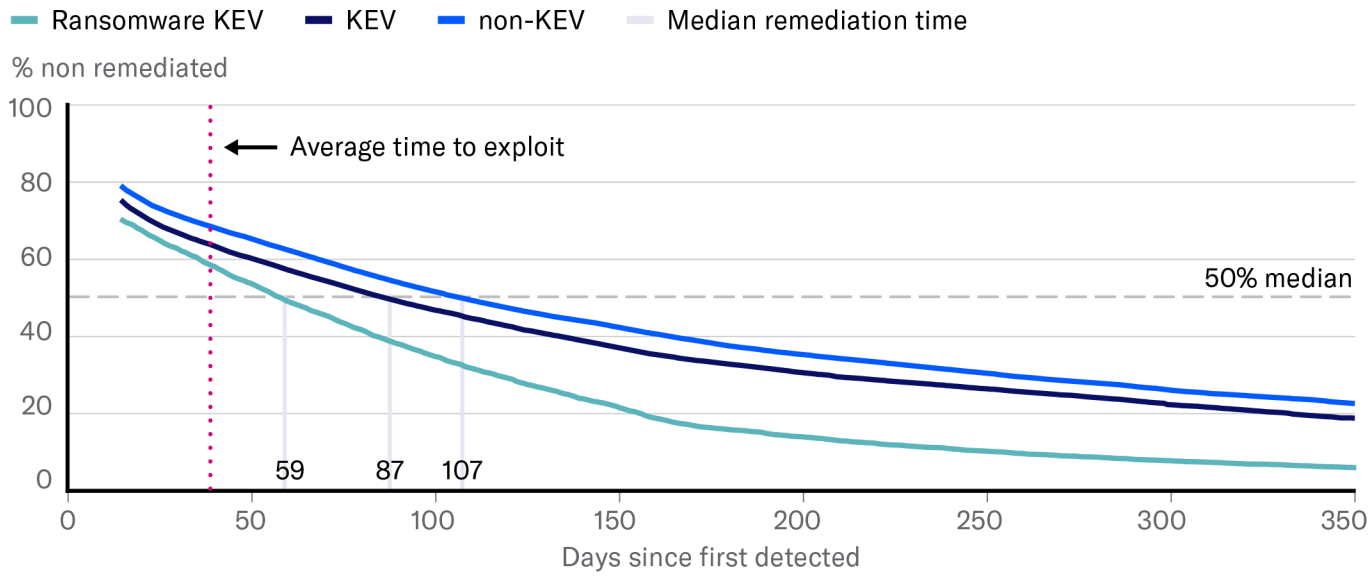
For all critical vulnerabilities, cataloged as Known Exploited Vulnerabilities (KEVs) by CISA, median remediation times extend to 87 days, or roughly three months. Less critical, uncataloged vulnerabilities (non-KEVs) take a median of 107 days to remediate, while data from cybersecurity analytics firm Bitsight show that the median time required to remediate top priority vulnerabilities—those that have been exploited by malicious actors to launch ransomware attacks (ransom KEVs)—is roughly 59 days, or nearly two months.

The exhibits below show how long it takes for vulnerabilities to disappear from the networks of organizations that Moody's rates. Exhibit 5 covers organizations from all sectors while Exhibit 6 breaks results down by sector.

Exhibit 5

Attackers exploit vulnerabilities more quickly than most organizations can fix them

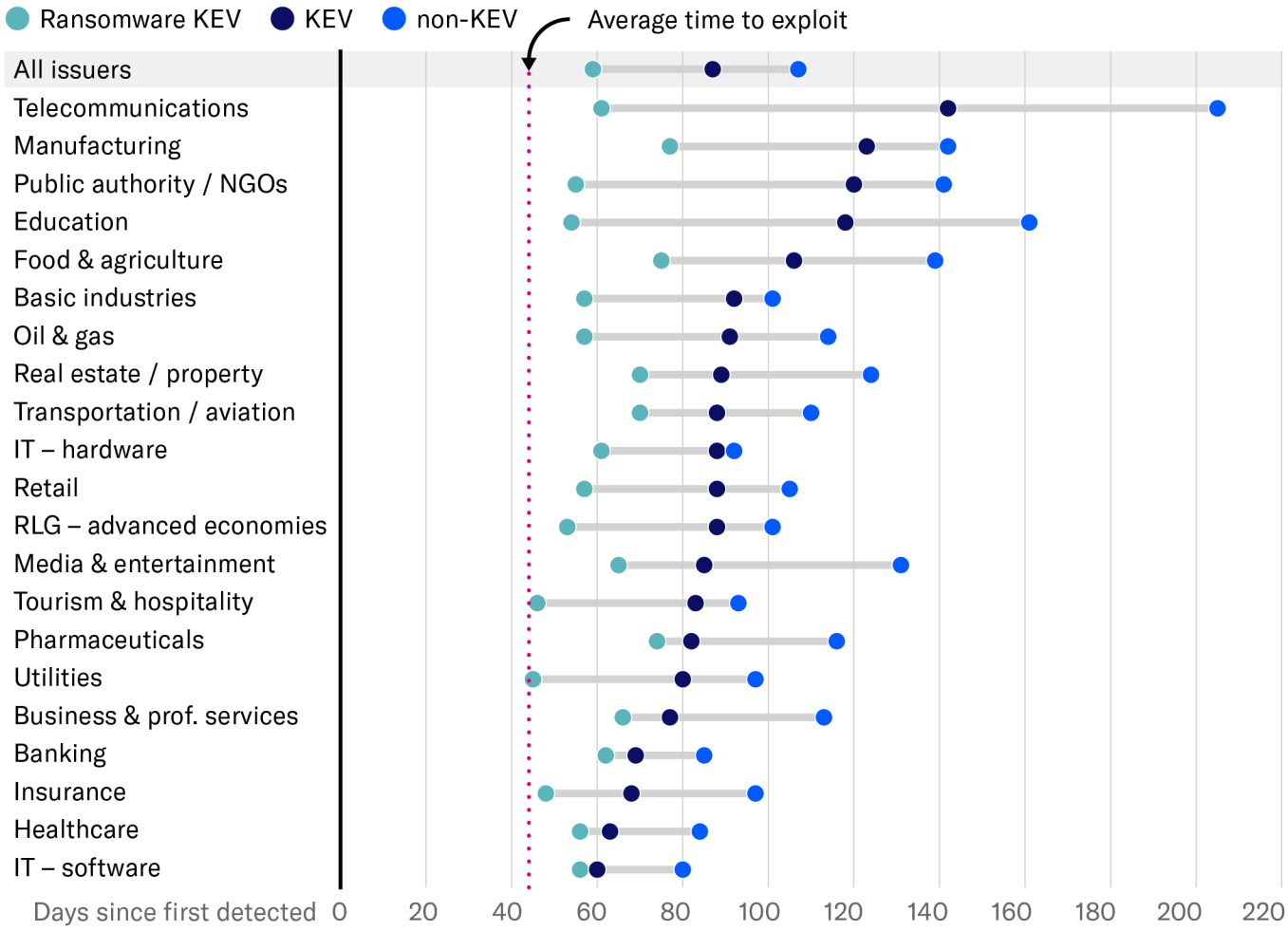
Survival curve of externally observable vulnerabilities, in days



Based on all vulnerabilities first detected in the last two years among 9,000+ Moody's-rated organizations

Sources: Bitsight, Moody's Ratings

Exhibit 6
Patching speed varies by sector
Median number of days to remediate vulnerabilities, by sector



Based on all vulnerabilities first detected in the last two years among 9,000+ Moody's-rated organizations
Source: Bitsight, Moody's Ratings

The time it takes attackers to exploit known vulnerabilities is shrinking

In a February report, cybersecurity researchers at Flashpoint reported that time-to-exploit, the average time between a vulnerability's public disclosure and its first known exploitation by attackers, fell sharply from 745 days in 2020 to just 44 days in 2025[2]. According to the researchers, this compression reflects a strategic shift among attackers, who are increasingly relying on publicly available vulnerability disclosures rather than developing bespoke exploit code.

These figures underscore a mismatch between attacker speed and defender response. While time-to-exploit is measured in weeks, time-to-remediation is measured in months. Defenders must carefully test and deploy fixes across complex environments; attackers, by contrast, face far fewer constraints and need only capitalize on a brief window of exposure. In this environment, vulnerabilities are left unpatched long after attackers are already capable of exploiting them, increasing the likelihood of operational disruption and financial loss.

The increasing use of AI to identify vulnerabilities in finished code will further exacerbate this challenge. While AI-driven discovery can improve visibility into software weaknesses, it also increases the volume of findings that security teams must assess and remediate. Without corresponding improvements in patching capacity, defender workloads are likely to grow at a time when the work of attackers is increasingly automated.

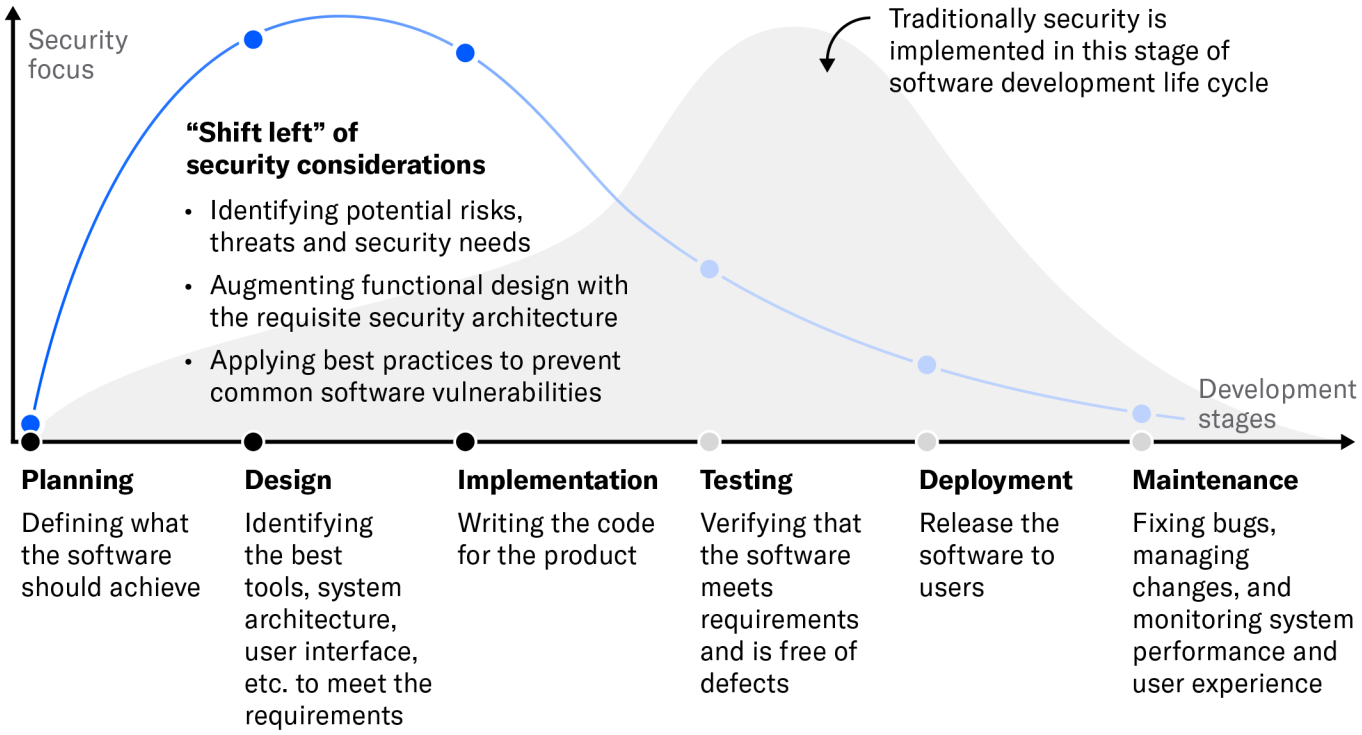
Secure coding practices will reduce the growing vulnerability backlog

The most effective way to manage the tremendous growth in vulnerability discovery is to shift detection earlier in the software development life cycle, where vulnerabilities can be addressed before products are released. In this context, secure-by-design software development represents the most durable approach to reducing risk for both organizations and their stakeholders.

Secure-by-design is a development practice that treats security as a core design requirement from the outset, rather than as a control added late in the process. While traditional software development typically addressed security in the final testing phase, this approach emphasizes anticipating, identifying, and mitigating potential weaknesses during design and development, before they become embedded in finished products (Exhibit 7). By reducing the number of vulnerabilities that reach production environments, secure-by-design practices help limit downstream remediation burdens and lower the likelihood that weaknesses are exploited in live systems.

Exhibit 7

An earlier focus on cybersecurity reduces the number of vulnerabilities in production software



Source: Moody's Ratings

AI is increasingly supporting this shift left in security. Integrated into modern software development platforms, AI-based tools can assist developers by reviewing code changes for security flaws as they are introduced. These tools can explain potential vulnerabilities in plain language, helping developers understand why a particular pattern is risky, and can suggest more secure alternatives aligned with established best practices. By embedding this feedback directly into routine development workflows, AI will enable faster, more consistent security decisions without requiring developers to be security specialists.

Endnotes

1 In a bug bounty program, an organization offers rewards (often monetary) to external security researchers for responsibly identifying and reporting vulnerabilities in its software or systems so they can be fixed before they are exploited by an attacker.

2 For publicly disclosed vulnerabilities

within the PRC for any regulatory purpose or for any other purpose which is not permitted under relevant PRC laws or regulations. For the purposes of this disclaimer, "PRC" refers to the mainland of the People's Republic of China, excluding Hong Kong, Macau and Taiwan.

REPORT NUMBER 1475622

CLIENT SERVICES

Americas	1-212-553-1653
Asia Pacific	852-3551-3077
Japan	81-3-5408-4100
EMEA	44-20-7772-5454